
Solving the Robust Matrix Completion Problem via a System of Nonlinear Equations

Yunfeng Cai and Ping Li

Cognitive Computing Lab

Baidu Research

No. 10 Xibeiwang East Road, Beijing 100085, China

10900 NE 8th St. Bellevue, WA 98004, USA

{caiyunfeng, liping11}@baidu.com

Abstract

We consider the problem of robust matrix completion, which aims to recover a low rank matrix L_* and a sparse matrix S_* from incomplete observations of their sum $M = L_* + S_* \in \mathbb{R}^{m \times n}$. Algorithmically, the robust matrix completion problem is transformed into a problem of solving a system of nonlinear equations, and the alternative direction method is then used to solve the nonlinear equations. In addition, the algorithm is highly parallelizable and suitable for large scale problems. Theoretically, we characterize the sufficient conditions for when L_* can be approximated by a low rank approximation of the observed M_* . And under proper assumptions, it is shown that the algorithm converges to the true solution linearly. Numerical simulations show that the simple method works as expected and is comparable with state-of-the-art methods.

1 Introduction

Robust matrix completion (RMC) (Chen et al., 2011; Tao and Yuan, 2011; Cherapanamjeri et al., 2017; Klopp et al., 2017; Zeng and So, 2018) aims to recover a low rank matrix L_* and a sparse matrix S_* from a sampling of $M = L_* + S_*$. Mathematically, RMC can be formulated as the following optimization problem (Candès et al., 2011; Chandrasekaran et al., 2011):

$$\begin{aligned} \text{RMC : } & \min_{L, S} \text{rank}(L) + \lambda |\text{supp}(S)|, \\ \text{s.t. } & L_{ij} + S_{ij} = M_{i,j}, \quad (i, j) \in \Omega, \end{aligned}$$

where λ is a tuning parameter, Ω is a subset of $\{1, \dots, m\} \times \{1, \dots, n\}$. When $S = 0$, RMC becomes the matrix completion (MC) problem (Candès and Recht, 2009; Meka et al., 2009; Cai et al., 2010; Candès and Tao, 2010; Jain and Netrapalli, 2015; Liu and Li, 2016); When $\Omega = \{1, \dots, m\} \times \{1, \dots, n\}$, RMC becomes the robust principal component analysis (RPCA) (Jolliffe, 2011). Thus, RMC can be taken as a combination/generalization of MC and RPCA (Candès and Plan, 2010; Jain and Netrapalli, 2015; Jain et al., 2013; Keshavan et al., 2010).

In many scientific and engineering problems, people need to recover a low rank matrix from observed data, e.g., the recommender system (Funk, 2006; Candès and Plan, 2010; Hu and Li, 2017, 2018b,a), social network analysis (Huang et al., 2013), machine learning (Candès and Plan, 2010; Davenport and Romberg, 2016), image inpainting (Bertalmio et al., 2000), computer vision (Candès and Plan, 2010), bioinformatics (Kim et al., 2005), etc.

MC/RPCA/RMC has been studied extensively from an optimization point of view, many algorithms are proposed, and exact recovery is discussed under proper assumptions. Most of well-known algorithms are based on convex optimization, in which the rank of a matrix is relaxed to its nuclear norm (the sum of all singular values), and the number of nonzero entries of a matrix is relaxed to its ℓ_1 -norm (the sum of absolute values of all entries), e.g., (Cai et al., 2010; Candès et al., 2011; Candès and Recht, 2009; Candès and Tao, 2010; Recht et al., 2010; Klopp et al., 2017). However, the computation of the nuclear norm of L , which requires the computation of its singular value decomposition (SVD), is expensive and unsuitable for parallelization, as a result, algorithms based on the nuclear norm relaxation are often not very realistic for large matrices.

To deal with large problems, the low rank matrix can be represented as the product of a tall-skinny matrix

and a short-fat matrix, so that the low rank property is satisfied automatically. But, the optimization problem becomes nonconvex, which makes it difficult to solve. Bi-convex as the problem is, the alternative minimization can be used to solve it more efficiently, e.g., (Jain et al., 2013). In (Yi et al., 2016), gradient descent method is used to solve RMC, which is shown to be fast. In (Cherapanamjeri et al., 2017), projected gradient method with hard-thresholding is used to solve RMC, with nearly-optimal observation and corruption. We refer the readers to (Zeng and So, 2018) and reference therein for more methods. Besides the optimization based method, quite recently, in (Dutta et al., 2019), the RPCA/RMC is solved via an alternating nonconvex projection method. This method does not require any objective function, convex relaxation or surrogate convex constraint.

Contribution. In this paper, we solve the RMC problem via solving a system of nonlinear equations (NLEQ). This method does not require any objective function, convex relaxation or surrogate convex constraint, either. Let $L_* = XY^T$, where $X = [x_1, \dots, x_m]^T \in \mathbb{R}^{m \times r}$, $Y = [y_1, \dots, y_n]^T \in \mathbb{R}^{n \times r}$. The RMC problem can be formulated as the following system of NLEQ:

$$x_i^T y_j = M_{ij}, \text{ for } (i, j) \in \Omega \setminus \text{supp}(S_*), \quad (1)$$

where $\text{supp}(S_*)$ is the support set of S_* . When the rank r and the support set of S_* are both known, solving RMC amounts solving a system of nonlinear equations. Any numerical method for nonlinear equations can be used to solve it (e.g., the steepest descent method, the Newton method, etc.), among which the simplest one is the alternative direction method (ADM): Fixing X (or Y), Y (or X) can be updated via solving an (overdetermined) linear system of equations (usually in least square sense). Thus, solving RMC via solving (1) heavily depends on whether we can solving (1) without knowing r and $\text{supp}(S_*)$. It is worth mentioning here that in (Meka et al., 2009), such a method was proposed to solve the MC problem. And it was stated there that “... its variants outperform most methods in practice. However, analyzing the performance of alternate minimization is a notoriously hard problem.”

The contributions of this paper are three folds. First, for both full and partial observation cases, we characterize some sufficient conditions for when the low rank approximation of the observed M is approximate L_* . Second, we propose to solve RMC via solving a system of NLEQ rather than optimization, and an ADM method, which carefully handles the unknown r and $\text{supp}(S_*)$ issue, is developed. Third, we carefully analyze the convergence of the ADM, and it is shown that under proper assumptions, the ADM converges to the true solution linearly, i.e., exact recovery can be

achieved. So, we give an answer to a problem which is even more difficult than the aforementioned “notoriously hard problem”. In addition, the algorithm is highly parallelizable and naturally suitable for large scale problems. It is also worth mentioning here that the results of this paper are applicable to the MC problem as well as the RPCA problem.

The rest of this paper is organized as follows. In Section 2, we first develop the algorithm, followed by its convergence analysis in Section 3. Numerical experiments are presented in Section 4. Concluding remarks are given in Section 5.

Notation. We shall adopt the MATLAB style convention to access the entries of vectors and matrices. The set of integers from i to j inclusive is $i : j$. For a matrix A , its submatrices $A_{(k:\ell, i:j)}$, $A_{(k:\ell, :)}$, $A_{(:, i:j)}$ consist of intersections of row k to row ℓ and column i to column j , row k to row ℓ and all columns, all rows and column i to column j , respectively. $A_{(j, :)}$ and $A_{(:, k)}$ denote the j th row and k th column of A , respectively. $\|A\|$ stands for the spectral norm of A , $\|A\|_F$ denotes the Frobenius norm, $\|A\|_1 = \sum_{i,j} |a_{ij}|$, $\|A\|_{\max} = \max_{i,j} |a_{ij}|$, $\|A\|_{2, \infty} = \max_i \|A_{(i, :)}\|$, A^\dagger stands for the Moore–Penrose inverse, and $\kappa(A) = \|A\| \|A^\dagger\|$ denote the condition number of A . Denote by $\sigma_j(A)$ for $1 \leq j \leq \min\{m, n\}$ the singular values of A and they are always arranged in a non-increasing order: $\sigma_1(A) \geq \sigma_2(A) \geq \dots \geq \sigma_{\min\{m, n\}}(A)$. $\mathcal{R}(A)$ stands for the range space of A , i.e., $\mathcal{R}(A) = \text{span}\{y \in \mathbb{R}^m \mid y = Ax, x \in \mathbb{R}^n\}$. The vector e_j stands for the j th column of the identity matrix I . Furthermore, for an index set $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$, $|\Omega|$ denotes the cardinality of Ω , $\Pi_\Omega(A) = [\mathcal{I}_{\{(i,j) \in \Omega\}} a_{ij}] \in \mathbb{R}^{m \times n}$, where \mathcal{I} is the indicator function.

2 Algorithm

In this section, we first reformulate the RMC problem as a problem of solving a system of nonlinear equations (NLEQ), show how to solve the NLEQ via an alternative direction method (ADM), then the overall algorithm is summarized.

2.1 Problem Reformulation, Difficulty and Solution

Let $M = XY^T$, where $X = [x_1, \dots, x_m]^T \in \mathbb{R}^{m \times r}$, $Y = [y_1, \dots, y_n]^T \in \mathbb{R}^{n \times r}$. The MC problem can be formulated as

$$x_i^T y_j = M_{ij}, \text{ for } (i, j) \in \Omega. \quad (2)$$

Now recall (1), the task of RMC becomes solving (2) with unknown r and minimum violators. Here by a “violator”, denoted by (i', j') , we mean that $x_{i'}^T y_{j'} \neq M_{i'j'}$, it will be also referred to as an “outlier” hereafter.

The ADM is the simplest method to solve the NLEQ of form (2) – given an initial guess for X , we fix X , (2) becomes a linear system in Y (which is assumed to be overdetermined), we can solve the linear system for Y ; similarly, we fix Y to solve X ; the iteration continues until convergence. As r and $\text{supp}(S)$ are unknown, our task is to determine them during the iteration of ADM. Next, we first show how to determine r , then $\text{supp}(S)$.

Determine the rank adaptively In the t th iteration of ADM, let the current rank estimation be r_t , Y_t be current estimation for Y and Y_t have orthonormal columns, i.e., $Y_t^T Y_t = I_{r_t}$. By ADM, the estimation of X can be obtained, denote it by \tilde{X}_{t+1} . In order to update r_t , we need to compute the singular values of $\tilde{X}_{t+1} Y_t^T$. Noticing that Y_t is orthonormal, then the top r_t singular values of $\tilde{X}_{t+1} Y_t^T$ will be the singular values of \tilde{X}_{t+1} . Then the singular values can be obtained as follows: first, compute the QR decomposition of \tilde{X}_{t+1} :

$$\tilde{X}_{t+1} = \hat{X}_{t+1} R_{x,t+1},$$

where $\hat{X}_{t+1} \in \mathbb{R}^{m \times r_t}$ is orthonormal, $R_{x,t+1} \in \mathbb{R}^{r_t \times r_t}$; second, compute the SVD of $R_{x,t+1}$:

$$R_{x,t+1} = Q_x \Sigma Q_y^T,$$

where $Q_x \in \mathbb{R}^{r_t \times r_t}$, $Q_y \in \mathbb{R}^{n \times r_t}$ are orthogonal, $\Sigma = \text{diag}(\hat{\sigma}_1, \dots, \hat{\sigma}_{r_t})$ with $\hat{\sigma}_1 \geq \dots \geq \hat{\sigma}_{r_t} \geq 0$. Then the singular values of $\tilde{X}_{t+1} Y_t^T$ are $\hat{\sigma}_1, \dots, \hat{\sigma}_{r_t}$. Similarly, when X_t is the current estimation for X , and $X_t^T X_t = I_{r_t}$, we can compute an estimation for Y , its singular values can be obtained.

Let L_t be the current estimation for L_* . When the rank is underestimated, i.e., $r_t < r$, the residual $\tau_t = \|\Pi_\Omega(L_t + S_t - M)\|_F$ will stagnate, in such case, we increase the estimated rank r_t . When the rank is overestimated, i.e., $r_t > r$, we expect to observe rank deficiency from the singular values $\hat{\sigma}_1, \dots, \hat{\sigma}_{r_t}$. In such a case, we decrease the estimated rank r_t . For the RMC problem, we prefer an overestimated rank over an underestimated rank due to the following reason. The residual τ_t stagnates for two reasons: one is that the estimated rank is smaller than the true rank; the other is that $|\text{supp}(S_*) \setminus \text{supp}(S_t)|$ is large. Then when the residual stagnates, it is difficult for us to make a good choice – to increase the estimated rank or to drop some equalities (of course, those equalities need to be carefully selected) in (2). Increasing the estimated rank when $|\text{supp}(S_*) \setminus \text{supp}(S_t)|$ is large or dropping equalities in (2) when the rank is underestimated will both lead to catastrophic consequences, such as the estimated rank exceeds a prescribed limit, too many “correct” equalities are dropped which will probably result in underdetermined linear systems when updating X (or Y). With an overestimated rank, when the

residual stagnates, we decrease the estimated rank via the singular values of L_t ; if there is no rank deficiency in L_t , we drop some equalities in (2).

When an overestimated rank decreases to the actual rank, it is expected that the estimated rank will remain unchanged in the follow-up iterations. Therefore, we do not need to check the singular values of L_t in each iteration for the sake of efficiency.

Determine $\text{supp}(S)$ via outlier detection When a good approximation \hat{L} of L_* is obtained, $S_* = M - L_* \approx M - \hat{L}$. Thus, it is reasonable to detect $(i, j) \in \Omega \cap \text{supp}(S_*)$ from the residual $\{R_{ij} = M_{ij} - \hat{L}_{ij}\}_{(i,j) \in \Omega}$.

Outlier detection has been used for centuries to remove abnormal data. Various outlier detection techniques have been used (Ester et al., 1996; Hodge and Austin, 2004; Xu et al., 2010; Rahmani and Li, 2019; Slawski et al., 2019). In our implementation, we simply determine the outliers as follows: find the top- k values in each row and column of $|R|$ (unavailable entries of $|R|$ are set to zero), and the entries in the intersection are taken as outliers. Alternatively, simply find the top k' values among all entries of $|R|$. Here k, k' are two parameters which can be tuned. In what follows, we denote

$$\mathcal{T}_s(A) = [b_{ij}], \quad (3)$$

where s is the number of the removed outliers, $b_{ij} = A_{(i,j)}$ if $A_{(i,j)}$ is an outlier, $b_{ij} = 0$, otherwise. Of course, one can also try other outlier detection techniques.

2.2 Algorithm details

Now we present Algorithm 1, which summarizes the ADM for RMC described in the previous subsection.

Some implementation details follows.

Initializing Y_0 According to Theorem 2 below, good initial guesses for X and Y can be obtained by computing the SVD of $\Pi_{\Omega_0}(M)$. An iterative procedure (e.g., Krylov subspace method) is usually adopted to accomplish the task, in which matrix vector products $\Pi_{\Omega_0}(M)v$ and $\Pi_{\Omega_0}(M)^T v$ are called several times. A simpler way, which is more efficient and numerically proven to be reliable, is the following: compute $W = \Pi_{\Omega_0}(M)^T \Pi_{\Omega_0}(M)N$, compute an orthonormal basis for W , and set the columns of Y_1 as the basis. Here $N \in \mathbb{R}^{n \times r_0}$ is a random matrix with entries drawn from the standard normal distribution. Such a procedure is essentially one iteration of the subspace method (a generalization of power method to compute several dominant eigenvectors). Since an initial guess for X or Y is sufficient for ADM to run in Algorithm 1, it is indeed unnecessary to compute the estimations for both X and Y .

Algorithm 1 ADM FOR RMC VIA NLEQ

Input: The observed matrix $\Pi_\Omega(M)$, a sparsity level parameter s , an estimated rank r_0 , an upper bound κ for the condition number of L_* , and a tolerance tol .

Output: $X \in \mathbb{R}^{m \times r_t}$, $Y \in \mathbb{R}^{n \times r_t}$ and $S \in \mathbb{R}^{m \times n}$ such that $\|\Pi_\Omega(XY^\top + S - M)\|_F \leq \text{tol}$, $\|S\|_0 \leq s$.

- 1: Set $S_0 = \mathcal{T}_s(M)$, $X_0 = 0$, $Y_0 = 0$, $\Sigma_0 = 0$, $t=1$;
 - 2: Compute $[X_1, \Sigma_1, Y_1] = \text{SVD}_{r_0}((M - S_0)/p')$, where $p' = (|\Omega| - s)/mn$;
 - 3: Compute $R_t = \Pi_\Omega(M - X_t \Sigma_t Y_t^\top)$;
 - 4: Set $S_t = \mathcal{T}_s(R_t)$, $\Omega_t = \Omega \setminus \text{supp}(S_t)$;
 - 5: Compute $\tau_t = \|\Pi_\Omega(M - X_t \Sigma_t Y_t^\top - S_t)\|_F$;
 - 6: **while** $\tau_t > \text{tol}$ **do**
 - 7: Set $t = t + 1$;
 - 8: Solve $\Pi_{\Omega_{t-1}}(\tilde{X}_t Y_{t-1}^\top) = \Pi_{\Omega_{t-1}}(M)$ for \tilde{X}_t ;
 - 9: Compute the QR decomposition $\tilde{X}_t = \tilde{X}_t R_{x,t}$, where \tilde{X}_t has orthonormal columns, $R_{x,t}$ is upper triangular;
 - 10: Compute the SVD $R_{x,t} = Q_x \hat{\Sigma} Q_y^\top$, where $\hat{\Sigma} = \text{diag}(\hat{\sigma}_1, \dots, \hat{\sigma}_{r_{t-1}})$, Q_x, Q_y are orthogonal;
 - 11: Set $r_t = r_{t-1} - |\{j \mid \kappa \hat{\sigma}_j < \hat{\sigma}_1\}|$;
 - 12: Set $\tilde{X}_t = [\tilde{X}_t Q_x]_{(:,1:r_t)}$;
 - 13: Solve $\Pi_{\Omega_{t-1}}(\tilde{X}_t \tilde{Y}_t^\top) = \Pi_{\Omega_{t-1}}(M)$ for \tilde{Y}_t ;
 - 14: Compute the QR decomposition $\tilde{Y}_t = \tilde{Y}_t R_{y,t}$, where \tilde{Y}_t has orthonormal columns, $R_{y,t}$ is upper triangular;
 - 15: Compute the SVD $R_{y,t}^\top = Q_x \hat{\Sigma} Q_y^\top$, where $\hat{\Sigma} = \text{diag}(\hat{\sigma}_1, \dots, \hat{\sigma}_{r_t})$, Q_x, Q_y are orthogonal;
 - 16: Set $r_t = r_t - |\{j \mid \kappa \hat{\sigma}_j < \hat{\sigma}_1\}|$;
 - 17: Set $X_t = [\tilde{X}_t Q_x]_{(:,1:r_t)}$, $Y_t = [\tilde{Y}_t Q_y]_{(:,1:r_t)}$, $\Sigma_t = \hat{\Sigma}_{(1:r_t, 1:r_t)}$;
 - 18: Compute $R_t = \Pi_\Omega(M - X_t \Sigma_t Y_t^\top)$;
 - 19: Set $S_t = \mathcal{T}_s(R_t)$, $\Omega_t = \Omega \setminus \text{supp}(S_t)$;
 - 20: Compute $\tau_t = \|\Pi_\Omega(M - X_t \Sigma_t Y_t^\top - S_t)\|_F$;
 - 21: **end while**
-

Solving X_t and Y_t On Lines 8 and 13, \tilde{X}_t and \tilde{Y}_t can both be solved row by row or simultaneously. And to obtain one row of \tilde{X}_t or \tilde{Y}_t , a small linear system needs to be solved. When the linear system is underdetermined, Algorithm 1 may break down. Therefore, in each row and column, Algorithm 1 requires the number of observed entries (after the removal of the corrupted entries) must be larger than the rank. To be more precise, we need the small linear system to be good conditioned. In general, it is difficult to determine how many rows/columns are needed to ensure the linear system to be good conditioned. Numerically, for a random matrix $A \in \mathbb{R}^{s \times r}$ (generated from a standard normal distribution) with $s = \mathcal{O}(r) > 2r$ is usually good conditioned. So, we may declare that $\mathcal{O}(r) > 2r$ observations in each row and column are sufficient.

In our implementation, the linear systems are solved in the least square sense. One may also choose to minimize ℓ_p -norm ($p \geq 0$) of the residual as in (Zeng and So, 2018).

Computational complexity When the number of observations in each row and column is $\mathcal{O}(r)$, each linear system can be solved in $\mathcal{O}(r^3)$ FLOPS. So, in each iteration, the computational complexity of the linear system solving on Lines 8 and 13 is $\mathcal{O}((m+n)r^3)$. The computational complexity of the QR decomposition on lines 9 and 14 is $\mathcal{O}((m+n)r^2)$. The computational complexity of the SVD is $\mathcal{O}(r^3)$. So, the overall of computational complexity of Algorithm 1 is dominated by the linear system solving. When the number of observations in certain row/column is much larger than r , we may randomly choose $\mathcal{O}(r)$ observations from the row/column, then solve a much smaller linear system of equations. Again, the overall computational complexity in each iteration is $\mathcal{O}((m+n)r^3)$.

Also, note that the linear systems on Line 8 and 13 can be solved in parallel. Therefore, Algorithm 1 are suitable for large scale problems.

Remark 1. When Ω_t is fixed, Algorithm 1 essentially minimizes $\|\Pi_{\Omega_t}(XY^\top - M)\|_F$ via ADM. If gradient method is used to minimize $\|\Pi_{\Omega_t}(XY^\top - M)\|_F$, Algorithm 1 is similar to the GD method in (Yi et al., 2016), except the regularization term $\|U_t^\top U_t - V_t^\top V_t\|_F$ in the loss function.

3 Convergence

This section analyzes the convergence of Algorithm 1. We first study the full observation case, which serves as a motivation for the partial observation case next.

To present the results, we need to define the k canonical angles. Let \mathcal{X}, \mathcal{Y} be two k -dimensional subspaces of \mathbb{R}^n . Let $X, Y \in \mathbb{R}^{n \times k}$ be the orthonormal basis matrices of \mathcal{X} and \mathcal{Y} , respectively, i.e.,

$$\mathcal{R}(X) = \mathcal{X}, \quad X^\top X = I_k, \quad \text{and} \quad \mathcal{R}(Y) = \mathcal{Y}, \quad Y^\top Y = I_k.$$

Denote ω_j for $1 \leq j \leq k$ the singular values of $Y^\top X$ in ascending order, i.e., $\omega_1 \leq \dots \leq \omega_k$. The k canonical angles $\theta_j(\mathcal{X}, \mathcal{Y})$ between \mathcal{X} and \mathcal{Y} are defined by

$$0 \leq \theta_j(\mathcal{X}, \mathcal{Y}) := \arccos \omega_j \leq \frac{\pi}{2} \quad \text{for } 1 \leq j \leq k. \quad (4)$$

They are in descending order, i.e., $\theta_1(\mathcal{X}, \mathcal{Y}) \geq \dots \geq \theta_k(\mathcal{X}, \mathcal{Y})$. Set

$$\Theta(\mathcal{X}, \mathcal{Y}) = \text{diag}(\theta_1(\mathcal{X}, \mathcal{Y}), \dots, \theta_k(\mathcal{X}, \mathcal{Y})). \quad (5)$$

In what follows, we sometimes place a vector or matrix in one or both arguments of $\theta_j(\cdot, \cdot)$ and $\Theta(\cdot, \cdot)$ with the meaning that it is about the subspace spanned by the vector or the columns of the matrix argument.

3.1 Full observation case

Theorem 1. Let $M = L_* + S_* \in \mathbb{R}^{m \times n}$ ($m \geq n$), where L_* is low rank, i.e., $r = \text{rank}(L_*) \ll n$. Let the SVD of M be $M = U\Sigma V^T$, where $U = [U_1 | U_2] = [u_1, \dots, u_r | u_{r+1}, \dots, u_m]$, $V = [V_1 | V_2] = [v_1, \dots, v_r | v_{r+1}, \dots, v_n]$ are orthogonal matrices, $\Sigma = \begin{bmatrix} \text{diag}(\Sigma_1, \Sigma_2) \\ 0 \end{bmatrix}$, $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$, $\Sigma_2 = \text{diag}(\sigma_{r+1}, \dots, \sigma_n)$, and $\sigma_1 \geq \dots \geq \sigma_n$. Let $M_r = U_1 \Sigma_1 V_1^T$ be the best rank r approximation of M . Let the economy sized SVD of L be $L_* = U_* \Sigma_* V_*^T$, where $U_* \in \mathbb{R}^{m \times r}$ and $V_* \in \mathbb{R}^{n \times r}$ both have orthonormal columns, $\Sigma_* = \text{diag}(\sigma_{1*}, \dots, \sigma_{r*})$ with $\sigma_{1*} \geq \dots \geq \sigma_{r*} > 0$. If

$$\|(I - U_* U_*^T) S_* (I - V_* V_*^T)\| < \sigma_{r*}, \quad (6a)$$

$$\max\{\|S_* V_*\|, \|S_*^T U_*\|\} < \sigma_r - \sigma_{r+1}, \quad (6b)$$

then

$$\begin{aligned} \max\{\theta_u, \theta_v\} &\leq \eta, \\ \frac{\|L_* - M_r\|_{\max}}{\|L_*\|} &\leq (\|U_*\|_{2,\infty} \theta_v + \|V_*\|_{2,\infty} \theta_u) \\ &\quad + (1 + 3\|U_*\|_{2,\infty} \|V_*\|_{2,\infty}) \theta_u \theta_v, \end{aligned}$$

where $\theta_u = \|\sin \Theta(U_1, U_*)\|$, $\theta_v = \|\sin \Theta(V_1, V_*)\|$ and $\eta = \frac{\max\{\|S_* V_*\|, \|S_*^T U_*\|\}}{\sigma_r - \sigma_{r+1} - \max\{\|S_* V_*\|, \|S_*^T U_*\|\}}$.

Theorem 1 tells that when (6) holds and η is small, the principal angles between U_1 and U_* , V_1 and V_* will be small, and the best rank- r approximation of M is a good approximation of L_* . Notice that, (6) does not necessarily implies $\|S_*\|$ is small (compared with $\|L_*\|$). In fact, we have the following example, in which $\|S_*\|$ is comparable with $\|L_*\|$ and $\eta = 0$.

Example 1. Let $M = L + S$, $L = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$,

$$S = \frac{\rho}{4} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & -1 & 2 \end{bmatrix}, \text{ where } \mathbf{1}_n \text{ is an } n\text{-by-1}$$

vector of ones, n is even, $\rho \in (-1, 1)$ is a real parameter. One can verify that $\|L\| = 1$, $\|S\| = \rho$, $\|S \mathbf{1}_n\| = 0$, the first two singular values of M are $\sigma_1 = 1$ and $\sigma_2 = |\rho|$, and the economy sized SVD of L can be given by $L = U_* \Sigma_* V_*^T$, where $U_* = V_* = \frac{1}{\sqrt{n}} \mathbf{1}_n$, $\Sigma_* = \sigma_{1*} = 1$. Then $\|(I - U_* U_*^T) S (I - V_* V_*^T)\| = \|S\| = |\rho| < 1 = \sigma_{1*}$, and $\max\{\|S V_*\|, \|S^T U_*\|\} = 0 < 1 - |\rho| = \sigma_1 - \sigma_2$. In other words, the assumption (6) holds. Noticing that $\eta = \frac{\max\{\|S V_*\|, \|S^T U_*\|\}}{\sigma_r - \sigma_{r+1} - \max\{\|S V_*\|, \|S^T U_*\|\}} = 0$, by Theorem 1, we can conclude that $\|\sin \Theta(U_1, U_*)\| = \|\sin \Theta(V_1, V_*)\| = 0$, $M_1 = L$, where U_1, V_1 are the top left and right

singular vectors of M , respectively, and M_1 is the best rank-1 approximation of M .

Now let us assume all entries of M are observed, S_* is sufficiently small, and can be taken as a perturbation to L_* . Let Y_t be guess for Y and have orthonormal columns, let us perform one iteration of ADM. For the ease of illustration, also assume that $r_{t+1} = r_t$. Then the iteration reads: **(S1)** $\tilde{X}_{t+1} = M Y_t$; **(S2)** $\hat{X}_{t+1} = M Y_t G_x$, where G_x is such that \hat{X}_{t+1} is orthonormal; **(S3)** $\tilde{Y}_{t+1} = M^T \hat{X}_{t+1}$; **(S4)** $\hat{Y}_{t+1} = M^T \hat{X}_{t+1} G_y$, where G_y is such that \hat{Y}_{t+1} is orthonormal. So, we get

$$\hat{Y}_{t+1} = (M^T M) Y_t G_x G_y, \quad (7)$$

which is just one iteration of subspace iteration (a generation of power iteration) for computing the dominant eigenspace of $M^T M$ (e.g., (Demmel, 1997; Stewart, 2001; Van Loan and Golub, 2012)). In fact, **(S2)** and **(S4)** are iterations for subspaces spanned by the dominant right and left singular vectors of M , respectively. Classic results tell that the subspaces $\mathcal{R}(X_t)$ and $\mathcal{R}(Y_t)$ converge to the subspaces spanned by the right and left singular vectors of M corresponding with the dominant singular values. When the perturbation is small, $\mathcal{R}(X_t)$ and $\mathcal{R}(Y_t)$ are good approximations for $\mathcal{R}(L_*)$ and $\mathcal{R}(L_*^T)$, respectively. In particular, when $S_* = 0$ and $V_*^T Y_t$ is nonsingular, we have $\|\sin \Theta(X_{t+1}, U_*)\| = 0$, $\|\sin \Theta(Y_{t+1}, V_*)\| = 0$, i.e., one iteration of ADM gives the true solution.

3.2 Partial observation case

For the partial observation case, we study the convergence of Algorithm 1 under the following assumptions:

(A1) For L_* , the column and row incoherence conditions with parameter μ hold, i.e.,

$$\max_{1 \leq i \leq m} \|U_*^T e_i\|^2 \leq \frac{\mu r}{m}, \quad \max_{1 \leq j \leq n} \|V_*^T e_j\|^2 \leq \frac{\mu r}{n}.$$

(A2) S_* has at most ρ -fraction nonzero entries per row and column, i.e.,

$$\|S_{*(i,:)}\|_0 \leq \rho n, \quad \|S_{*(:,j)}\|_0 \leq \rho m, \quad \text{for all } i, j.$$

(A3) Each entry of M is observed independently with probability p .

Besides the notations in Algorithm 1, we also adopt the following notations:

$$\theta_{x,t} = \|\sin \Theta(X_t, U_*)\|, \quad \theta_{y,t} = \|\sin \Theta(Y_t, V_*)\|.$$

The following theorem tells that the SVD of the partial observed matrix (after the removal of outlier) indeed

gives good approximation for U_*, V_* . Furthermore, X_1, Y_1 satisfy a incoherence condition with parameter μ_1 , $\|L_* - X_1 \Sigma_1 Y_1^T\|_{\max}$ is bounded.

Theorem 2. Assume (A1), (A2), (A3) and $m \geq n$. Let $M = L_* + S_* \in \mathbb{R}^{m \times n}$ with $\text{rank}(L_*) = r$, S_0 be obtained as in Algorithm 1. Denote $r'_s = \frac{\|S_0 - S_*\|_F^2}{\|S_0 - S_*\|_F^2}$, $\gamma = \frac{2}{1-\varrho} \sqrt{\frac{2\varrho}{r'_s p}}$. If

$$(\xi + \gamma)\mu r \kappa < \frac{1}{6}, \quad (8)$$

then with probability $\geq 1 - 1/m^{10+\log \alpha}$, it holds that

$$\max\{\theta_{x,1}, \theta_{y,1}\} \leq 3(\xi + \gamma)\mu r \kappa, \quad (9)$$

where $\xi = 6\sqrt{\frac{\alpha}{p'n}}$, $\kappa = \frac{\sigma_{1*}}{\sigma_{r*}}$. Further assume $\mu \ll n$ and that there exists a positive constant $\mu'_1 \ll n$ such that

$$(\xi + \gamma)\mu r \kappa \leq \frac{1}{3}\sqrt{\frac{\mu'_1 r}{m}}, \quad (10)$$

then

$$\begin{aligned} \|X_1\|_{2,\infty} &\leq \sqrt{\frac{\mu_1 r}{m}}, & \|Y_1\|_{2,\infty} &\leq \sqrt{\frac{\mu_1 r}{n}}, \\ \|L_* - X_1 \Sigma_1 Y_1^T\|_{\max} &\leq \|L_*\| \left(\sqrt{\frac{\mu r}{m}} \theta_{y,1} + \sqrt{\frac{\mu r}{n}} \theta_{x,1} \right. \\ &\quad \left. + \theta_{x,1} \theta_{y,1} \right) + \mathcal{O}(n^{-3/2}), \end{aligned}$$

where $\mu_1 = 2(\mu + \mu'_1)$.

Remark 2. When there is no corruption, i.e., $\varrho = 0$, then $\gamma = 0$. Furthermore, when $m = \mathcal{O}(n) \gg 1$, since $p'n = \mathcal{O}(1)r$, we know that $\xi = \sqrt{\frac{\alpha}{p'n}}$ is small, the larger $p'n$ is, the smaller ξ is. By Theorem 1, $\theta_{x,1}$ and $\theta_{y,1}$ will be small. In other words, the SVD $\frac{1}{p'} \Pi_{\Omega_0}(M - S_0) = X_1 \Sigma_1 Y_1^T$, gives good approximation for both U_* and V_* , by X_1 and Y_1 , respectively.

The following theorem, which is motivated by (Drineas and Mahoney, 2018, Lemma 55), establish the bridge between the full observation case and the partial observation case. This gives an upper bound for the distance between the least square solutions between the full observation case and the partial observation case.

Theorem 3. Let $m \geq n$, and denote

$$\begin{aligned} X_{\text{opt}} &= \operatorname{argmin}_X \|XY_t^T - (M - S_t)\|, \\ \tilde{X}_{\text{opt}} &= \operatorname{argmin}_X \|\Pi_{\Omega_t}(XY_t^T - (M - S_t))\|. \end{aligned}$$

Assume that Ω_t can be obtained by sampling each entry of M with probability $p' = p(1 - \varrho)$, $\|Y_t\|_{2,\infty} \leq \sqrt{\frac{\mu'r}{n}}$

for some $\mu' > 0$, and

$$\inf_{X \in \mathbb{R}^{m \times r}} \frac{\|\Pi_{\Omega_t}(XY_t^T)\|}{\|X\|} \geq \sigma \quad (11)$$

for some constant $\sigma > 0$. Then w.p. ≥ 0.99 , it holds

$$\|\tilde{X}_{\text{opt}} - X_{\text{opt}}\| \leq \left(\frac{2}{3} \log(m+n) + 5\right) \frac{\sqrt{\mu'r p'}}{\sigma^2} \|R\|_{\max},$$

where $R = (M - S_t)(I - Y_t Y_t^T)$.

Remark 3. The requirement (11) is critical. The parameter σ reflects the condition number of the least square problem on line 8 of Algorithm 1. What's more, the larger p' is, the larger σ is (in particular, if $p' = 1$, $\sigma = 1$), the smaller the distance between \tilde{X}_{opt} and X_{opt} is, which agrees with our intuition. R is the residual for the full observation case, i.e., $R = X_{\text{opt}} Y_t^T - (M - S_t)$. If the residual is small, the distance between \tilde{X}_{opt} and X_{opt} will be small, too.

Definition 1. Define $\mu' \triangleq \max\{\mu_u, \mu_v\}$, where

$$\begin{aligned} \mu_u &\triangleq \sup_{U \in \mathbb{R}^{m \times r}} \left\{ \frac{m}{r} \|U\|_{2,\infty}^2 \mid \|\sin \Theta(U_*, U)\| \leq \theta_{x,1} \right\}, \\ \mu_v &\triangleq \sup_{V \in \mathbb{R}^{n \times r}} \left\{ \frac{n}{r} \|V\|_{2,\infty}^2 \mid \|\sin \Theta(U_*, V)\| \leq \theta_{y,1} \right\}. \end{aligned}$$

By definition of μ' , we know that if $\theta_{x,t} \leq \theta_{x,1}$, $\theta_{y,t} \leq \theta_{y,1}$ for all t , then X_t, Y_t satisfy the incoherence condition with parameter μ' . Recall Theorem 1, under the assumption of (10), $\theta_{x,1}$ and $\theta_{y,1}$ are quite small (at the order of $\frac{1}{\sqrt{m}}$), then we can show that $\mu' \leq \mu_1$, which implies that μ' is not large.

The next theorem establishes the convergence rate for the ADM, which is the key in our proof of Theorem 5.

Theorem 4. Assume that Ω_t can be obtained by sampling each entry of M with probability p' , $\|Y_t\|_{2,\infty} \leq \sqrt{\frac{\mu'r}{n}}$ for some $\mu' > 0$, (11) and

$$\|L_* - X_t \Sigma_t Y_t^T\|_{\max} \leq c \|L_*\| \theta_{y,t} \sqrt{\frac{\mu r}{m}}, \quad (12)$$

for some constants $c > 0$. Denote

$$\begin{aligned} r_s &= \inf_t \frac{\|S_t - S_*\|_F^2}{\|S_t - S_*\|^2}, & \zeta &= \sqrt{\frac{2s\mu r}{mr_s}}, \\ C_{\text{LS}} &= \left(\frac{2}{3} \log(m+n) + 5\right) \frac{\sqrt{\mu'r p'}}{\sigma^2}, \\ C &= C_{\text{LS}}(1 + 2c\sqrt{2p\varrho n})\sqrt{\mu r}, \\ \epsilon &= c\kappa\zeta, & \phi &= \frac{8\epsilon(\kappa + \sqrt{2\epsilon}) + \sqrt{2}C\kappa/\sqrt{m}}{1 - 2\epsilon - C\kappa/\sqrt{m}}. \end{aligned}$$

Further assume $\theta_{y,t} \leq \frac{1}{\sqrt{2}}$, then w.p. ≥ 0.99 ,

$$\theta_{x,t+1} \leq \phi \theta_{y,t}.$$

Remark 4. The assumption (12) is not a strong requirement as it looks. By Theorem 1, (12) is natural for $t = 1$. For general $t > 1$, it can be shown that there exists a constant $c > 0$ such that (12) holds (see supplementary for details), as long as X_t, Y_t satisfy the incoherence condition. In general, the constant c is at the order of $\mathcal{O}(1)$.

Remark 5. The constant r_s is the infimum of the stable rank of $S_t - S_*$. If we take a random matrix, whose entries are i.i.d. drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$, to approximate $S_t - S_*$. Numerically, we found that $r_s = \mathcal{O}(n)$. Therefore, $\epsilon \approx c\kappa\sqrt{2\rho\mu r}$ is a small number as long as ρ is small, μ and κ are not large. When p' is sufficiently large, $C = \mathcal{O}(1)$. Consequently, when $m = \mathcal{O}(n)$ is large, $\phi \ll 1$, which agrees with our conclusion in the full observation case.

Remark 6. Under similar assumptions as in Theorem 4, it can also be shown that $\theta_{y,t+1} \leq \phi\theta_{x,t+1}$. Then $\theta_{y,t+1} \leq \phi^2\theta_{y,t}$. When $\phi < 1$, $\{\theta_{y,t}\}_t$ is a monotonically decreasing sequence. Combining it with the definition of μ' , we know that Y_t satisfies the incoherence condition with parameter μ' . Similarly, X_t also satisfies the incoherence condition.

Theorem 5. Assume (A1), (A2), (A3) and $m \geq n$. Assume that Ω_t can be obtained by sampling each entry of M with probability $p' = p(1 - \rho)$, $r_t \equiv r$ and

$$\inf_{X \in \mathbb{R}^{m \times r}} \frac{\|\Pi_{\Omega_t}(XY_t^T)\|}{\|X\|} \geq \sigma, \quad \inf_{Y \in \mathbb{R}^{n \times r}} \frac{\|\Pi_{\Omega_t}(\hat{X}Y^T)\|}{\|Y\|} \geq \sigma$$

for some $\sigma > 0$. Let r_s, ζ, C, ϵ be the same as in Theorem 4. Then with high probability, it holds that

$$\|M - S_{t+1} - X_{t+1}\Sigma_{t+1}Y_{t+1}^T\| \leq \psi \|M - S_t - X_t\Sigma_tY_t^T\|,$$

where

$$\psi = \frac{2\sqrt{2}(\kappa + 2\epsilon\sqrt{\frac{\mu r}{m}} + \frac{C\kappa}{\sqrt{m}})(8\epsilon(\kappa + \sqrt{2}\epsilon) + \sqrt{2}\frac{C\kappa}{\sqrt{m}})}{(1 - 4\sqrt{2}\epsilon\sqrt{\frac{\mu r}{m}})(1 - 2\epsilon - \frac{C\kappa}{\sqrt{m}})}.$$

Remark 7. If $\psi < 1$, then by Theorem 5, $\{\|M - S_t - X_t\Sigma_tY_t^T\|\}_t$ is a monotonically decreasing sequence. And in limit, with high probability, it holds

$$\lim_{t \rightarrow \infty} \|M - S_t - X_t\Sigma_tY_t^T\| = 0.$$

Recall the way we determine S_t , we get $(L_* - X_t\Sigma_tY_t^T)_{ij} = 0$ for any $(i, j) \notin \text{supp}(S_t)$. Then we can show that

$$X_t\Sigma_tY_t^T \rightarrow L_*, \quad S_t \rightarrow S_*, \quad \text{as } t \rightarrow \infty,$$

i.e., exact recovery is achieved, with high probability.

4 Experiments

We compare Algorithm 1 (NLEQ) with the gradient descent (GD) method (Yi et al., 2016) and the PG-RMC method (Cherapanamjeri et al., 2017). The codes of GD and PG-RMC are obtained from the lrslibrary (Sobral et al., 2016) in Github.¹

4.1 Synthetic Data

We generate the data matrix $M \in \mathbb{R}^{d \times d}$ as follows. The low rank matrix L_* is given by $L_* = U_*V_*^T$, where the entries of $U_*, V_* \in \mathbb{R}^{d \times r}$ are drawn independently from the Gaussian distribution with mean zero, variance $1/d$. Each entry of the sparse matrix S are nonzero with probability ρ , and the nonzero entries of S are uniformly drawn from $[-\frac{r}{2d}, \frac{r}{2d}]$. Each entry of $M = L_* + S_*$ is observed independently with probability p .

The results are presented in Figure 1. In Figure 1-(a), we draw the relative residual $\frac{\|R_t\|_F}{\|\Pi_{\Omega_t}(M)\|_F}$ and rank estimation r_t vs. iteration number. We can see that when the initial rank is larger than the true rank, as the iteration continues, the true rank can be revealed from the singular values of $X_t(Y_t)^T$, then the rank estimation drops to the true rank; when the residual stagnates (represented by a big solid dot in the plot), outliers are removed and the residual decreases until convergence. In Figure 1-(b), we plot the relative residual $\frac{\|R_t\|_F}{\|\Pi_{\Omega_t}(M)\|_F}$ vs. total CPU time for different p . We can see that Algorithm 1 works for all three cases, the larger p is, the less iteration number is needed. In Figure 1-(c), we plot total CPU time and the angle $\max\{\|\sin \Theta(X_t, U_*)\|, \|\sin \Theta(Y_t, V_*)\|\}$ vs. matrix size d for different methods. We can see that the CPU time of all three methods grows linearly with respect to the matrix size, and are comparable with each other. The angles of the three methods are all small, which confirm that all methods give the correct results; the angle produced by NLEQ is the smallest. In Figure 1-(d), we plot the relative residual $\frac{\|R_t\|_F}{\|\mathcal{P}_{\Omega_t}(M)\|}$ vs. CPU time for all three methods. The convergence behaviors of three methods are quite different: GD converges almost linearly; PG-RMC at the beginning stage converges linearly with a low converge rate, then converges almost linearly with a larger rate; NLEQ has a zig-zag convergence, which is due to the removal of outliers.

4.2 Foreground-background separation

The next task is foreground-background separation. By stacking up the vectorized video frames, we get a full data matrix. The static background will form a low rank matrix while the foreground can be taken as

¹<https://github.com/andrewssobral/lrslibrary/tree/master/algorithms/mc>

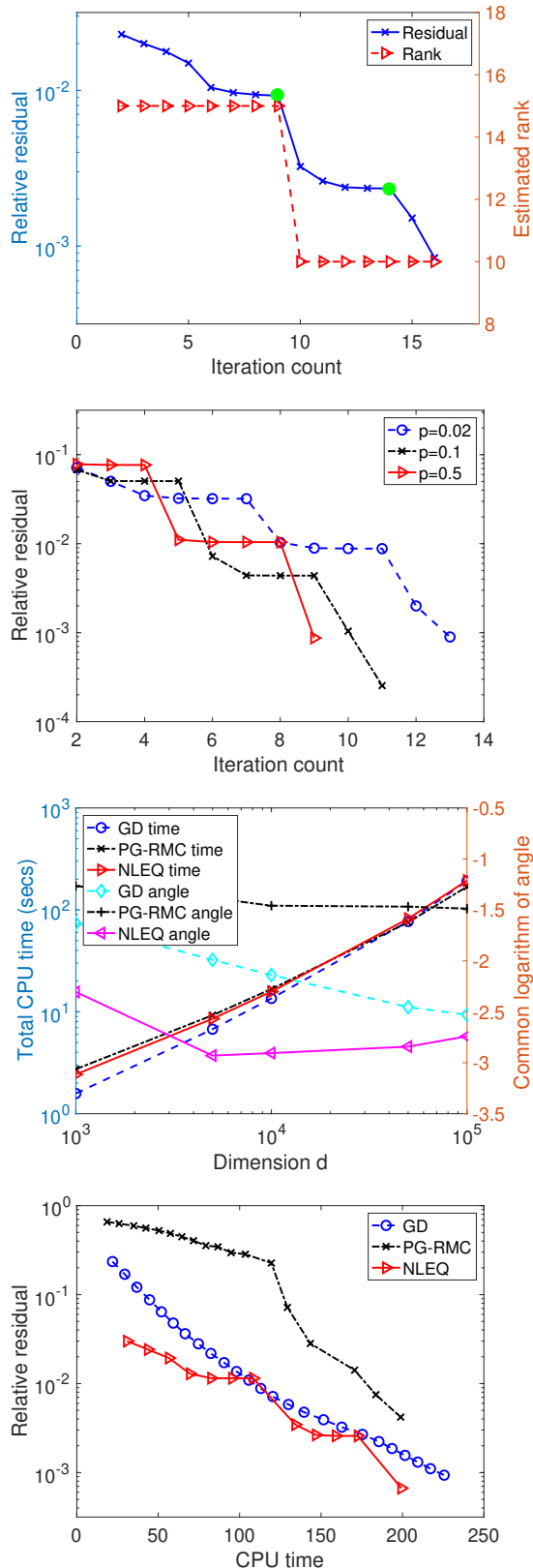


Figure 1: Results on synthetic data. Up to down: (a)-(d). (a) $d = 1e5$, $r = 10$, $p = 0.0015$, $\rho = 0.1$. (b) $d = 1e4$, $r = 10$, $\rho = 0.1$, $p = 0.02, 0.1, 0.5$. (c) $d = 1e3, 5e3, 1e4, 5e4, 1e5$, $r = 10$, $p = 0.15r^2 \log(m)/m$, $\rho = 0.1$. (d) $d = 1e5$, $r = 10$, $p = 0.002$, $\rho = 0.1$.

the sparse component. We apply our method NLEQ, and also GD and PG-RMC to two public benchmarks, the *Bootstrap* and *ShoppingMall*.² Each entry of the data matrix is observed independently w.p. $p = 0.05$. As presented in Figure 2, all three methods are able to separate the foreground from the background, and the backgrounds obtained by three methods are similar.

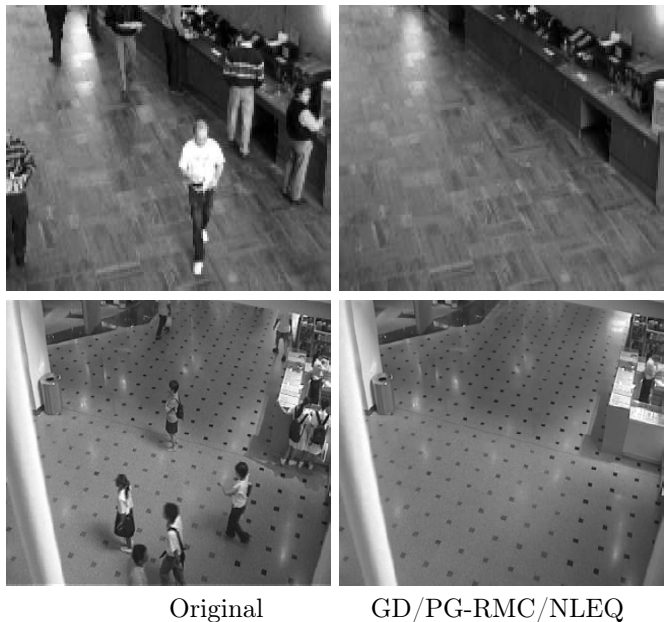


Figure 2: Foreground-background separation.

5 Conclusion

In this paper, we study the RMC problem from an algebraic point of view – transform the RMC problem into a problem of solving an overdetermined nonlinear system of equations (with outliers). This method does not require any objective function, convex relaxation or surrogate convex constraint. Algorithmically, we propose to solve the NLEQ via ADM, in which the true rank and support set of the corruption are determined during the iteration. The algorithm is highly parallelizable and suitable for large scale problems. Theoretically, we characterize the sufficient conditions for when L_* can be approximated by the low rank approximation of M or $\frac{1}{p}\Pi_{\Omega}(M)$. We establish sufficient conditions for $M_r \approx L_*$, where M_r is the best rank r approximation of the observed M . The convergence of the algorithm is guaranteed, and exact recovery is achieved under proper assumptions. Numerical simulations show that the algorithm is comparable with state-of-the-art methods in terms of efficiency and accuracy.

²<http://vis-www.cs.umass.edu/~narayana/castanza/I2Rdataset/>

References

- Marcelo Bertalmío, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 417–424, New Orleans, LA, 2000.
- Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010.
- Emmanuel J. Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
- Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717, 2009.
- Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Information Theory*, 56(5):2053–2080, 2010.
- Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011.
- Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.*, 21(2):572–596, 2011.
- Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust matrix completion and corrupted columns. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 873–880, Bellevue, WA, 2011.
- Yeshwanth Cherapanamjeri, Kartik Gupta, and Prateek Jain. Nearly optimal robust matrix completion. In *Proceedings of the 34th International Conference on Machine Learning, (ICML)*, pages 797–805, Sydney, Australia, 2017.
- Mark A. Davenport and Justin K. Romberg. An overview of low-rank matrix recovery from incomplete observations. *J. Sel. Topics Signal Processing*, 10(4):608–622, 2016.
- James W Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.
- Petros Drineas and Michael W Mahoney. Lectures on randomized numerical linear algebra. *The Mathematics of Data*, 25:1, 2018.
- Aritra Dutta, Filip Hanzely, and Peter Richtárik. A nonconvex projection method for robust PCA. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pages 1468–1476, Honolulu, HI, 2019.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, Portland, OR, 1996.
- Simon Funk. Netflix update: Try this at home, 2006.
- Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.
- Jun Hu and Ping Li. Decoupled collaborative ranking. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1321–1329, Perth, Australia, 2017.
- Jun Hu and Ping Li. Collaborative multi-objective ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1363–1372, Torino, Italy, 2018a.
- Jun Hu and Ping Li. Collaborative filtering via additive ordinal regression. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*, pages 243–251, Marina Del Rey, CA, 2018b.
- Jin Huang, Feiping Nie, Heng Huang, Yu Lei, and Chris H. Q. Ding. Social trust prediction using rank-k matrix recovery. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2647–2653, Beijing, China, 2013.
- Prateek Jain and Praneeth Netrapalli. Fast exact matrix completion with finite samples. In *Proceedings of The 28th Conference on Learning Theory (COLT)*, pages 1007–1034, Paris, France, 2015.
- Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Symposium on Theory of Computing Conference (STOC)*, pages 665–674, Palo Alto, CA, 2013.
- Ian Jolliffe. *Principal component analysis*. Springer, 2011.
- Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Trans. Information Theory*, 56(6):2980–2998, 2010.
- Hyunsoo Kim, Gene H. Golub, and Haesun Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, 2005.
- Olga Klopp, Karim Lounici, and Alexandre B Tsybakov. Robust matrix completion. *Probability Theory and Related Fields*, 169(1-2):523–564, 2017.
- Guangcan Liu and Ping Li. Low-rank matrix completion in the presence of high coherence. *IEEE Trans. Signal Processing*, 64(21):5623–5633, 2016.

- Raghu Meka, Prateek Jain, and Inderjit S. Dhillon. Matrix completion from power-law distributed samples. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1258–1266, Vancouver, Canada, 2009.
- Mostafa Rahmani and Ping Li. Outlier detection and robust PCA using a convex measure of innovation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 14200–14210, Vancouver, Canada, 2019.
- Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.
- Martin Slawski, Mostafa Rahmani, and Ping Li. A sparse representation-based approach to linear regression with partially shuffled labels. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, page 7, Tel Aviv, Israel, 2019.
- Andrews Sobral, Thierry Bouwmans, and El-hadi Zahzah. Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos. *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, 2016.
- Gilbert W Stewart. *Matrix algorithms volume 2: eigen-systems*, volume 2. SIAM, 2001.
- Min Tao and Xiaoming Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM J. Optim.*, 21(1):57–81, 2011.
- Charles F Van Loan and Gene H Golub. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2012.
- Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2496–2504, Vancouver, Canada, 2010.
- Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4152–4160, Barcelona, Spain, 2016.
- Wen-Jun Zeng and Hing-Cheung So. Outlier-robust matrix completion via p -minimization. *IEEE Trans. Signal Processing*, 66(5):1125–1140, 2018.